UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/595,003	06/13/2000	Nicolas Vazquez	5150-44300	7955

7590 12/12/2007

Jeffrey C. Hood
Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767

EXAMINER	
PILLAI, NAMITHA	
ART UNIT	**PAPER NUMBER**
2173	

MAIL DATE	DELIVERY MODE
12/12/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/595,003
Filing Date: June 13, 2000
Appellant(s): VAZQUEZ ET AL.

09/595, 003
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 9/13/07 appealing from the Office action mailed 3/22/07.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,862,372	MORRIS	1-1999
0510514 A1	OKA	10-1992

4,831,580	YAMADA	5-1989
5,623,659	SHI	4-1997

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-16, 21-37, 39-43, 45-59, 61-65, 67-74 and 76-90 are rejected under 35 U.S.C. 103(a) as being unpatentable over U. S. Patent No. 5,862,372 (Morris et al.), herein referred to as Morris, EP Publication No. 0510514 A1 (Oka et al.), herein referred to as Oka and U. S. Patent No. 4,831,580 (Yamada).

Referring to claims 1, 53, 71, 81 and 90, Morris discloses a method for creating a graphical program to perform an algorithm, with the recording of functions in response to user input, with the functions specifying the algorithm (column 2, lines 20-27). Morris discloses automatically generating the graphical program in response to the recorded functions, with the display of a plurality of interconnected nodes, which visually indicate functionality of the graphical program, with the graphical program implementing the algorithm. See column 3, lines 29-35. Morris does not disclose that user input does not use the selection of the nodes. Oka discloses automatically generating a graphical program represented as the flowchart with the program functions, where the blocks or

nodes representing each function is automatically generated, selected and formed into a graphical program without any user intervention with the graphical program being automatically generated (column 1, lines 20-26). It would have been obvious for one skilled in the art, at the time of the invention to learn from Oka to implement an automatic process in which both the blocks and the connections between the blocks are automatically generated to implement a program. The systems and objectives of creating a graphical program are similar in both Morris and Oka. Both inventions describe a process for generating a graphical program, with the nodes of the program representing functions. Oka alleviates user interaction by further teaching automatically generating of the graphical flowchart without any user intervention. The graphical program of Oka is generated by automatically generating the blocks or nodes representing the functions and automatically connecting the nodes to generate a representation of the graphical program. Hence, one skilled in the art would have been motivated to learn from Oka to automatically generate the flowchart, through including of the nodes without direct user input selecting the nodes.

Morris and Oka do not disclose that the automatically created flowchart is generated based on an algorithm, where the flowchart generation involves creation of a program. Yamada discloses a flowchart which when created, generates a program (column 2, lines 8-27). It would have been obvious to one skilled in the art at the time of the invention to learn from Yamada that the flowchart generated is a program that is created for execution from the generated flowchart. The combination of Morris and Oka have taught that a flowchart can automatically be generated with Yamada further teaching the

generated flowcharts can represent program that are generated when the flowchart is created for automatic generation of a program. The combination of Morris and Oka could further automate a previously manual process by relying on Yamada to learn that an automatically generated flowchart can be used in automatic creation of a graphical program. Therefore, one skilled in the art would have been motivated to learn from Yamada to that the flowchart generated creates a program automatically.

Referring to claims 2, 54, 79 and 88, Morris discloses performing the function in response to user input, wherein the user dragging the objects is the function which is then recorded in response to the user's action for specifying the algorithm (column 3, lines 32-34).

Referring to claims 3, 32 and 55, Morris discloses creating a prototype or a set of instructions by recording the functions carried out by the user (column 3, lines 29-30).

Referring to claims 4, 33, 56, 72 and 82, Morris discloses that the prototype is in the discipline of image processing as seen by the results of the prototyping program shown in Figure 5 (column 6, lines 37-42).

Referring to claims 5, 34 and 57, Morris discloses recording the functions in response to input received via a graphical user interface (column 2, lines 20-24 and Figure 5).

Referring to claims 6, 58, 73 and 83, Morris discloses that the graphical user interface wherein the user would work with is based on the prototyping environment application (column 2, lines 57-52).

Referring to claims 7, 35, 59, 74 and 84, Morris discloses that the user input consists of selecting the functions from a menu and palette (column 2, lines 20-24).

Referring to claims 9, 39, 61, 76 and 85, Oka discloses automatically generating the graphical program comprises automatically including and connecting the nodes generating graphical code in the graphical program without direct user input (column 1, lines 20-26).

Referring to claims 10 and 62, Morris discloses the graphical program running or executing, wherein the algorithm represented by the functions are performed (column 3, lines 34-35).

Referring to claims 11, 40, 63 and 87, Morris discloses that the graphical program includes a block diagram portion (column 5, line 43) and a user interface panel portion, represented as the palette in Figure 5.

Referring to claims 12, 41 and 78, Morris discloses that the graphical program is a graphical data flow program as seen in Figure 2 (column 5, lines 43-46).

Referring to claims 13, 42, 64, 77 and 86, Oka discloses that the automatic generation of the graphical program is done through the automatic inclusion of nodes or "objects" corresponding to respective one of the one or more functions in the graphical program (column 1, lines 20-26).

Referring to claims 14, 43 and 65, Morris discloses the functions comprising a script, with the script having an association with the graphical program (column 6, lines 24-26). Morris discloses modifying the script to create a new script in response to user input

once an association has been made (column 6, lines 26-33). Morris also discloses modifying the graphical program, shown as the map view of objects for the graphical program, and this map representation being modified based on the changes to the script, with the production of the new script (column 6, lines 32-43).

Referring to claim 15, Morris discloses there being a clear association between the script and the graphical program, this association being used during modification of the graphical program and this association remaining between the new script and the new graphical program (column 5, lines 54-55 and column 6, lines 24-36).

Referring to claim 16, Morris discloses receiving user input indicating a desire to change the graphical program, displaying the script information of the script, modifying the script information in response to user input and modifying the graphical program after modifying the script information. See column 6, lines 9-14).

Referring to claims 21, 45 and 67, Morris discloses receiving user input specifying code generation information and using this information to automatically generate the graphical program (column 2, lines 20-27).

Referring to claims 22, 46 and 68, Morris discloses that the code generation information represented as objects, specify or represent the type of program to create in response to the recorded function, wherein the program is created in accordance with the specified graphical program type (column 3, lines 5-15).

Referring to claims 23 and 47, Morris does disclose that the graphical program type, in this case being "WINDOWS" applications, specifies a particular programming environment, wherein the program, represented as "APPLICATIONS" are created in a

file format that is usable by the particular programming environment. Morris clearly discloses using standard programming languages, representing the file format, which would enable the users to create the applications particular to the programming environment being used. See column 4, lines 45-55.

Referring to claims 24, 48 and 69, Morris discloses a plurality of parameters associated with the functions wherein each parameter is an input parameter, which provides input to a function (column 6, lines 26-29). Morris also discloses that the code generation information specifies input parameters, which are desired to be interactively changeable. Morris also discloses automatically generating the graphical program represented by objects, wherein the program receives user input during the program operation, with the user specifying values for the specified input parameters. See column 6, lines 24-32. Morris also discloses automatically generating the graphical program comprises enabling the graphical program to display output during program operation, wherein the output indicates values for the specified parameters (column 6, lines 37-41).

Referring to claims 25, 49, 70, 80 and 89, Morris discloses automatically generating a graphical program includes generating portions of graphical code, with each portion implementing one of the functions and linking the portions of graphical code together (column 1, lines 61-66 and column 2, lines 1-4).

Referring to claims 26 and 50, as seen in Figure 2 of Morris, the graphical programming nodes each have inputs and outputs, and wherein generating the portions of the graphical code comprises connecting the node inputs and outputs together in

order to implement the function with which the portion of graphical code is associated (column 5, lines 40-50).

Referring to claims 27 and 51, as seen in Figure 2 of Morris, a first portion of graphical code is linked to a second portion of graphical by connecting an output of a node in the first portion to an input of a node in the second portion of the graphical code.

Referring to claim 28, Morris discloses data being passed between the objects, wherein the data affecting the outcome of these objects (column 5, lines 46-48). As seen in Figure 2 also, the functions represented as the objects have input parameters, wherein the portion of code with the node has an input for receiving a value for the input parameter. Each of the nodes having an input parameter also has a leaf node that has an output for providing a value for the input parameter, with the leaf node for providing the parameter value is connected to the node input for receiving the parameter value as seen by the node relationships shown in Figure 2.

Referring to claim 29, Morris discloses that the functions have output parameters, with as seen in Figure 2, certain nodes providing output parameters to other node, to implement functions, wherein there is a leaf node that is associated with the node with the output parameter, with the leaf node receiving the output parameter as input for the node, with the two nodes being connected to each other. See Figure 2.

Referring to claims 30 and 52, Morris discloses that al l information used for this invention, which would include the information needed to generate the graphical program and the functionalities of the nodes are all stored remote information source or

storage means, thereby suggesting a database, as shown in Figure 1 (column 5, lines 10-16).

Referring to claim 31, Morris discloses a computer system, which would include a processor with a memory, coupled to the processor, which would store certain applications, one of them being a prototyping environment application (column 5, lines 8-11). Morris also discloses a user input device which receives user input, the presence of a user input obviously being inherently disclosed through the discussion of the user manipulating the icons from the palette (column 2, line 24). Morris discloses that this prototyping environment application carries out the generation of the graphical program (column 1, lines 10-15). Morris discloses a method for creating a graphical program to perform an algorithm, with the recording of functions in response to user input, with the functions specifying the algorithm (column 2, lines 20-27). Morris discloses automatically generating the graphical program in response to the recorded functions, with the display of a plurality of interconnected nodes, which visually indicate functionality of the graphical program, with the graphical program implementing the algorithm. See column 3, lines 29-35. Morris does not disclose that user input does not use the selection of the nodes. Oka discloses automatically generating a graphical program represented as the flowchart with the program functions, where the blocks or nodes representing each function is automatically generated, selected and formed into a graphical program without any user intervention with the graphical program being automatically generated (column 1, lines 20-26). It would have been obvious for one skilled in the art, at the time of the invention to learn from Oka to implement an

automatic process in which both the blocks and the connections between the blocks are automatically generated to implement a program. The systems and objectives of creating a graphical program are similar in both Morris and Oka. Both inventions describe a process for generating a graphical program, with the nodes of the program representing functions. Oka alleviates user interaction by further teaching automatically generating of the graphical flowchart without any user intervention. The graphical flowchart of Oka is generated by automatically generating the blocks or nodes representing the functions and automatically connecting the nodes to create a flowchart. Hence, one skilled in the art would have been motivated to learn from Oka to automatically generate the flowchart, through including of the nodes without direct user input selecting the nodes.

Morris and Oka do not disclose that the automatically created flowchart is generated based on an algorithm, where the flowchart generation involves creation of a program. Yamada discloses a flowchart which when created, generates a program (column 2, lines 8-27). It would have been obvious to one skilled in the art at the time of the invention to learn from Yamada that the flowchart generated is a program that is created for execution from the generated flowchart. The combination of Morris and Oka have taught that a flowchart can automatically be generated with Yamada further teaching the generated flowcharts can represent program that are generated when the flowchart is created for automatic generation of a program. The combination of Morris and Oka could further automate a previously manual process by relying on Yamada to learn that an automatically generated flowchart can be used in automatic creation of a

graphical program. Therefore, one skilled in the art would have been motivated to learn from Yamada to that the flowchart generated creates a program automatically.

Referring to claim 36, Morris discloses a computer system implementing this graphical program creation program, wherein the information would be stored in memory and there would be means for calling the graphical program creation program, within the computer system, as would be the case for calling any application in a computer system (column 5, lines 8-13). Morris also discloses that the prototyping environment application through calling the run time program, whereby calling the graphical program creation program, executes to automatically generate the graphical program (column 7, lines 4-7).

Referring to claim 37, Morris discloses that the graphical creation program is a graphical programming development environment application (column 4, lines 46-50).

Claims 17-20, 44 and 66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morris, Oka and Yamada, and further in view of U. S. Patent No. 5,623,659 (Shi et al.).

Referring to claims 17, 44 and 66, Morris and Oka discloses creating an association between the script and the graphical program (Morris, column 6, lines 30-32). Morris, Oka and Yamada do not disclose locking the association between the script and the general program. Shi discloses locking the association between a program and a user, wherein the locking prevents other users from editing the portion associated to the initial user (column 2, lines 5-11). It would have been obvious to modify Morris, Oka and Yamada's invention such that locking the association between

the script and the graphical program wherein the locking prevents the user from editing the program. Morris discloses allowing the user's to manipulate the scripts and the graphical program, but does not disclose any means for controlling this manipulation, thereby possibly leading to mistakes. Thus measures must be taken to ensure that unnecessary mistakes do not occur. The locking mechanism disclosed by Shi could prevent such mistakes from occurring in Morris, Oka and Yamada's graphical program, and laying out some control mechanisms for user manipulation. One skilled in the art would be motivated to learn from Shi's teachings by locking the association between the script and the graphical program to have more control over the user's manipulations of the graphical program.

Referring to claim 18, Shi discloses unlocking the association between the script and the graphical program in response to user input after locking. Shi also discloses directly changing the portion in response to the changes made by the user. See column 2, lines 19-23.

Referring to claim 19, Shi discloses unlocking such that it removes the association between the script and the program (column 2, lines 19-20).

Referring to claim 20, Morris discloses modifying the graphical program in response to user input after generating the graphical program and after creating the association between the script and the graphical program (column 6, lines 20-36). Morris also discloses determining if an association exists between the script and the graphical program in response to the modifications made by the user concerning

components of the graphical program (column 6, lines 30-32). Morris, Oka and Yamada do not disclose removing the association between the script and the graphical program. Shi discloses removing the association between the script and the graphical program in response to modifying (column 2, lines 19-20). It would have been obvious for one skilled in the art, at the time of the invention to remove the association between the script and the graphical program in response to modifying. Morris, Oka and Yamada's invention denotes an association between the script and the graphical program at all times, which could prove inconvenient, especially when the user may make mistakes by changing the script and the association causing the same changes to the graphical program which may be undesirable. To prevent such unnecessary mistakes, Morris, Oka and Yamada could use a mechanism for removing the association between the script and the graphical program in response to modifying. One skilled in the art, at the time of the invention, would have been motivated to learn from Shi to implement a means for removing the association between the script and the graphical program in response to modifying.

(10) Response to Argument

Claims 1, 5, 7, 9, 10, 13, 39, 42, 53, 57, 59, 61, 62, 64, 76, 85 and 90

The graphical program as defined by Appellant's disclosure includes nodes that are connected together to carry out distinct functions through the execution of program data. Each of the nodes within this graphical program represents one function that is included in the entire graphical program. The prototyping environment allows for the automatic generation of a graphical program including the generation of text based

programs (page 8, lines 1-7). The graphical program that is generated is associated with a graphical programming development environment in which this graphical program is created.

Appellant's disclosure also discloses that U.S. Patent 4,901,221 also describes a graphical program environment where the graphical program is conveyed as a block diagram that is created graphically. The graphical program carries out a desired function by the user with appropriate input variables that provide the necessary data to carry out the function which results in the generation of an output. The instructions that are generated from this graphical program are compiled and interpreted to carry out the function where the graphical program must be compiled to a second code format of executable code instructions in order for the computer system to process the data to generate a desired output. See U.S. Patent 4,901,221, column 3, lines 50-65.

Appellant has argued that Morris does not teach automatic creation of a graphical program. The rejection states that it is the combination of Morris, Oka and Yamada that discloses the features of the present invention's claims. Based on the disclosure of these prior art references, the combination of Morris, Oka and Yamada are obvious and disclose the features of Appellant's claims.

Appellant argues that Oka discloses automatically generating a flowchart and not a graphical program. The disclosure of Oka provides motivation and suggests to a user the automatic generation of a program including automatic generation of nodes and connection of these nodes. Morris upon learning from Oka would be motivated to automatically generate and connect the nodes to further alleviate user reliance for

generating the graphical program which is a clear objective of Morris. The graphical program of Morris already discloses nodes, connecting the nodes and executable format for running the graphical program where in addition to these features Morris would be motivated to learn from Oka that such structure with nodes and node connections can be further automatically generated where the nodes and connections between the nodes are automatically created. The flowchart program of Oka and the graphical program of Morris both represent program data represented and conveyed as nodes with connections between the nodes disclosing the flow of the program. The teachings of Oka suggests to the user of Morris that the nodes and the connection between the nodes can be further automated in a system such as Morris' who's intentions are clearly to automate the program generation process. Oka's disclosure suggests to one of ordinary skill in the art that program data represented through nodes and connection between nodes can be automatically generated where these nodes and connection between the nodes are automatically made. The flowchart of Oka is clearly disclosing a program and program data that is represented through nodes and connections between these nodes.

Appellant argues that Yamada does not disclose graphical programs. The disclosure of Yamada discloses that flowcharts are executable and can generate executable program data. The flowchart which is clearly an executable program itself represents a graphical program with nodes and connections between these nodes. The combination of Morris, Oka and Yamada disclose the automatic generation of a graphical program with the nodes and node connections automatically being generated.

Appellant argues that Morris does not disclose recording functions in response to user input. The user selection of nodes and functions and storing of these selections as a graphical program reads on recording functions in response to user input where the user input is selection of these nodes and other functions in the graphical programming environment. Clearly user input and interaction is involved with the programming environment and the selections made by the user are associated with the graphical program including selecting the nodes and also adding input variables to customize the node functions. All of this data is recorded.

Appellant argues that there is no motivation to combine Morris, Oka and Yamada. Morris has disclosed the objective of automating a program generation process which provides motivation for Morris to further learn from Oka to automate the generation of the graphical program with the nodes and the connections between the nodes. Oka has disclosed the inconvenience of relying on user input to place nodes and node connections, where the automatic process of Oka would alleviate user interaction in program generation of Morris.

Claims 2 and 54

The user dragging and dropping is a process which is interpreted as a function. The nodes each represent functions. In response to user input, the process of dragging and dropping can be carried out. The dragging and dropping has to be recorded into the graphical program in order to generate the program and execute it.

Claims 11, 40, and 63

Figure 5 discloses a graphical programming environment with block diagram portion which displays a block diagram including nodes of a graphical program. The map view window discloses nodes with connections between the nodes which represents a block diagram. The palette clearly is a user interface panel portion. Furthermore, the graphical program has been represented through the flow diagram of the map window which contains block diagram portions with connections between each other. These nodes also represent user interface panel portions. The user interface components represented as nodes make the graphical programs and reads on block diagrams and user interface panels.

Claims 12 and 41

The graphical program generated with nodes and connections between the nodes represents a graphical data flow diagram where the graphical data includes nodes which flow through the diagram to carry out the functionality of the program.

Claims 14, 43 and 65

When the user had dragged and dropped a new node, this is user input where when the underlying script is generated as a result of this, there is clearly an association between this user interaction, the node and the underlying script. The addition of this new node along with the clearly new associated script data, there is the creation of a new script. The addition of this new node, with the new underlying script data also clearly changes the graphical program. Therefore, in response to this user input, a new graphical program and a new script are both generated.

Claim 15

Morris has disclosed that the graphical programs have scripts that are associated with it. This association between the nodes and the script must remain where the script defines the functionality of a specific node, therein the purpose for the association between a script and a node. The graphical program includes the new node that has been added by the user and the new script generated that is associated with the node. This addition of the node has clearly changed the node and generates a new graphical program based on the addition of the new node and therefore also the addition of the associated script data. It is because of this association between the new node and the script data that a new graphical program with new underlying script data is generated.

Claim 16

Modifying the script can be carried out by user input where user manipulation of the nodes in the graphical program can modify the script including addition and deletion of nodes where the script associated with those nodes would be modified and affecting the script associated with the graphical program. The script, the nodes associated with the scripts and the graphical program with the nodes are clearly associated with each other, where changes to the script would clearly affect the graphical program and the underlying script that is represented through all the nodes of the graphical program.

Claim 21, 45, and 67

The user selection of the node with the generation of the script associated with it reads on user input specifying code generation information. Based on the user's selections code data is generated that is associated with the node that is selected. This

code generation affects the graphical program to which the node has been added or created for.

Claims 22, 23, 46, 47 and 68

The graphical program of Morris and the associated code generated is a result of user input. The graphical program that is generated is based on the how the user has placed the nodes and the association between the nodes. Therefore one graphical program is different from another based on the user input and placement and connections of the nodes in addition to the further input variables that are associated with each of the nodes. This results in different types of graphical programs.

Claims 24, 48 and 49

The program operation includes any operation of the program from interacting with the program code, running the program, modifying the program and executing the program. Therefore one skilled in the art at the time of the invention would not interpret a broad term such as program operation to specifically read only as execution of the program after it is created.

Claims 25, 49, 70, 80, 89

The graphical program of Morris is automatically generated in that the code and the association between the code and the nodes in the graphical program are automatically generated. The underlying code data that is automatically generated in Morris is linked together based on the structure of the graphical program. Clearly the objective of Morris is automatic code generation based on the structure of a graphical flow diagram.

Claims 26, 27-29, 50 and 51

The graphical code that is generated is associated with the nodes and the connections between the nodes. The functions of the program are implemented because of the nodes and the connection linking the nodes together. The nodes of the graphical program in Morris are clearly connected in order to generate the code.

Claims 30 and 52

The mass storage describes a database which stores data required for the system to carry out the functionality. The data stored in the database is used to generate a graphical program.

Claims 3, 6, 31, 32, 34, 35, 36, 37, 55, 58, 71, 73, 74, 77, 79, 81, 83, 84, 86, 87 and 88

Appellant has argued that Morris does not teach automatic creation of a graphical program. The rejection states that it is the combination of Morris, Oka and Yamada that discloses the features of the present invention's claims. Based on the disclosure of these prior art references, the combination of Morris, Oka and Yamada are obvious and disclose the features of Appellant's claims.

Appellant argues that Oka discloses automatically generating a flowchart and not a graphical program. The disclosure of Oka provides motivation and suggests to a user the automatic generation of a program including automatic generation of nodes and connection of these nodes. Morris upon learning from Oka would be motivated to automatically generate and connect the nodes to further alleviating user reliance for generating the graphical program which is a clear objective of Morris. The graphical program of Morris already discloses nodes, connecting the nodes and executable

format for running the graphical program where in addition to these features Morris would be motivated to learn from Oka that such structure with nodes and node connections can be further automatically generated where the nodes and connections between the nodes are automatically created. The flowchart program of Oka and the graphical program of Morris both represent program data represented and conveyed as nodes with connections between the nodes disclosing the flow of the program. The teachings of Oka suggests to the user of Morris that the nodes and the connection between the nodes can be further automated in a system such as Morris who's intentions are clearly to automate the program generation process. Oka's disclosure suggests to one of ordinary skill in the art that program data represented through nodes and connection between nodes can be automatically generated where these nodes and connection between the nodes are automatically made. The flowchart of Oka is clearly disclosing a program and program data that is represented through nodes and connections between these nodes.

Appellant argues that Yamada does not disclose graphical programs. The disclosure of Yamada discloses that flowcharts are executable and can generate executable program data. The flowchart which is clearly an executable program itself represents a graphical program with nodes and connections between these nodes. The combination of Morris, Oka and Yamada disclose the automatic generation of a graphical program with the nodes and node connections automatically being generated.

Appellant argues that Morris does not disclose recording functions in response to user input. The user selection of nodes and functions and storing of these selections as

a graphical program reads on recording functions in response to user input where the user input is selection of these nodes and other functions in the graphical programming environment. Clearly user input and interaction is involved with the programming environment and the selections made by the user are associated with the graphical program including selecting the nodes and also adding input variables to customize the node functions. All of this data is recorded.

Appellant argues that there is no motivation to combine Morris, Oka and Yamada. Morris has disclosed the objective of automating a program generation process which provides motivation for Morris to further learn from Oka to automate the generation of the graphical program with the nodes and the connections between the nodes. Oka has disclosed the inconvenience of relying on user input to place nodes and node connections, where the automatic process of Oka would alleviate user interaction in program generation of Morris.

A prototyping environment is one where prototype or set of data can be created. In this case, Morris discloses an environment in which a prototype as a graphical program can be created. This prototype defines how distinct functionality can be carried out when the program is executed.

Claims 4, 33, 72, 82, 56

The output of the program includes image data where the graphical program including image processing to generate this output of an image in response to execution of the graphical program.

Claims 78, 87

The graphical program with the nodes and the connections between the nodes represent a diagram model of the algorithm that is generated. This algorithm is implemented and executable as a graphical program in Morris.

Claims 17-20, 44, and 66

Shi does disclose preventing a user from editing the graphical program through a locking means. With Shi disclosing that at least one user is prevented from editing program data, this reads on preventing or managing user editing of the graphical program. The point of locking is to prevent a user from editing a certain portion of data which Shi has successfully done by locking a certain portion to ensure that a certain user would not have access to and editing means for changing data.

An objective of Morris is to alleviate user interaction where the code that is being generated including script code is based on simply interacting with a graphical program structure and not always with the script code data which is automatically generated. Therefore, there is a motivation for unnecessary user editing of script and program data where the automatic generation of code is created for such prevention of user interaction. Therefore, a locking means which locks the generated code data with the graphical program would ensure that user interaction with the code is not occurring as is an intention of Morris' system. Therefore the combination of Morris, Oka, Yamada and Shi is obvious because a person of ordinary skill has a good reason to pursue these known locking options which are within the technical grasp of one skilled in the art at the time of the invention. It is clear that these known techniques would lead to anticipated success of alleviating unnecessary user interaction with the code.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Namitha Pillai
Patent Examiner
Art Unit 2173
December 6, 2007

Conferees:

Lynne H. Browne
Appeal Practice Specialist, TQAS
Technology Center 2100
December 6, 2007

TADESSE HAILU
PRIMARY EXAMINER
Tadesse Hailu
Primary Examiner
Art Unit 2173
December 6, 2007